LABORATORIO 2

Unidad 3

Camila Paladines

Computación Científica

Profesor: Hernán Darío Vargas Cardona, PhD

Abril 9 de 2021

RESUMEN

En este informe se describen los resultados conseguidos al implementar algoritmos en Python para la obtención de respuestas a problemas matemáticos, como lo es el cómputo de los parámetros que proporcionan un ajuste lineal para un conjunto de datos, en este caso, mediante el Método de Ecuaciones Normales y el Método de Transformaciones Householder. Además, se realizan algunos análisis sobre los métodos utilizados aplicándolos a conjuntos de datos reales, donde se observa su comportamiento dependiendo de diferentes factores. También se calcula la complejidad computacional y la exactitud, con el fin de comparar los métodos y determinar cuál obtuvo mejores resultados para los conjuntos de datos usados.

ABSTRACT

This report describes the results obtained by implementing algorithms in Python to obtain answers to mathematical problems, such as the computation of the parameters that provide a linear fit to a data set, in this case, using the Normal Equations Method and the Householder Transformations Method. In addition, some analyses are performed on the methods used by applying them to real data sets, where their behavior is observed depending on different factors. The computational complexity and accuracy are also calculated, in order to compare the methods and determine which one obtained better results for the data sets used.

Contenido

1. Introducción **1**

2. Materiales y métodos **2**

2.1. Materiales 2

2.2. Métodos 2

2.2.1. Ajuste de Datos 2

2.2.2. Mínimos Cuadrados Lineal 2

2.2.3. Método de Ecuaciones Normales 3

2.2.4. Método de Transformaciones Householder 3

3. Resultados de las simulaciones **5**

3.1. Ejemplo 1 5

3.1.1. Parámetros x 5

3.1.2. Tiempo de cómputo 6

3.1.3. Exactitud de los algoritmos 7

3.2. Ejemplo 2 9

3.2.1. Parámetros x 9

3.2.2. Tiempo de cómputo 10

3.2.3. Exactitud de los algoritmos 11

4. Discusión y análisis **13**

4.1. Resultados de los métodos 13

4.2. Límite de los algoritmos 13

4.3. Complejidad computacional 13

4.4. Mejores valores de n 14

5. Conclusiones **15**

6. Referencias **16**

1. Introducción

En el mundo físico que conocemos existen algunos elementos o comportamientos que siguen ciertos patrones. Algunos de ellos son, por ejemplo, la sucesión de Fibonacci, presente en el crecimiento de la población de conejos, la cantidad de pétalos en algunas flores, etc. Desde el punto de vista matemático, se puede *predecir* el valor de salida para un valor de entrada (o un conjunto de valores), dado un patrón que se haya encontrado en ellos. Para realizar esta tarea se han planteado diferentes métodos, en este caso, se usarán el Método de Ecuaciones Normales y el Método de Transformaciones Householder para obtener los parámetros de una función y usarla para predecir los datos de salida dependiendo de un valor de entrada.

Estas herramientas se pueden aplicar a diferentes campos, siendo el aprendizaje automático el más destacado entre ellos, donde uno de sus objetivos principales es entrenar un modelo de predicción con un conjunto de entrenamiento y probar su exactitud con otro conjunto de validación que permita verificar que este modelo realmente aprendió.

2. Materiales y métodos

2.1. Materiales

Para el desarrollo de esta unidad se usó Python 3.7, con las librerías `numpy`, `matplotlib.pyplot`, `time` y `pandas`.

2.2. Métodos

2.2.1. Ajuste de Datos

Dados m datos $(t_i,\, y_i)$, el objetivo es encontrar un vector de parámetros x que provea el mejor ajuste a una función $f(t,\, x)$, donde $f : \mathbb{R}^{n+1} \to \mathbb{R}$. Este ajuste se obtiene mediante la siguiente expresión:

$$min\, x \sum_{i=1}^{m} (y_i - f(t_i, x))^2$$

conocida como la solución de mínimos cuadrados, que es lineal si la función f es lineal en los componentes del vector x. Entonces f es una combinación lineal de funciones ϕ que dependen de t:

$$f(t, x) = x_1\phi_1(t) + x_2\phi_2(t) + \cdots + x_n\phi_n(t)$$

Un ajuste lineal polinómico se puede expresar de la siguiente manera:

$$f(t, x) = x_1 + x_2 t + x_3 t^2 + \cdots + x_n t^{n-1}$$

2.2.2. Mínimos Cuadrados Lineal

Un problema de mínimos cuadrados lineal se puede escribir en notación matricial de la siguiente manera:

$$Ax \approx b$$

Donde $A_{ij} = \phi_j(t_i)$ y $b_i = y_i$. Por ejemplo, en el ajuste polinomial cuadrático el cual tiene tres parámetros ($f(t, x) = x_1 + x_2 t + x_3 t^2$), y teniendo un conjunto de

5 datos $(t_1, y_1), \ldots, (t_5, y_5)$, entonces la matriz A sería de 5×3, y el problema se podría representar así:

$$Ax = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_1^2 \\ 1 & t_3 & t_1^3 \\ 1 & t_4 & t_1^4 \\ 1 & t_5 & t_1^5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = b$$

2.2.3. Método de Ecuaciones Normales

Un sistema de ecuaciones normales de $n \times n$ dado por $A^\top Ax = A^\top b$ se puede usar para solucionar el problema de mínimos cuadrados lineal $Ax \approx b$.

La matriz $A^\top A$ es simétrica y positiva definida, por lo cual se puede computar su descomposición de Cholesky:

$$A^\top A = LL^\top$$

donde L es una matriz triangular inferior y $L^\top$ triangular superior. La solución x del problema de mínimos cuadrados se puede computar solucionando dos sistemas triangulares: $Ly = A^\top b$ y $L^\top x = y$.

2.2.4. Método de Transformaciones Householder

Una transformación Householder es una matriz de la forma:

$$H = I - 2\frac{vv^\top}{v^\top v}$$

donde v es un vector sin ceros. Una matriz $H = H^\top = H^{-1}$ es ortogonal y simétrica. Dado un vector a, se busca encontrar un vector v tal que:

$$Ha = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha e_1$$

El vector v en este caso se puede hallar mediante:

$$v = a - \alpha e_1$$

tomando $\alpha = \pm \|a\|_2 = \sqrt{a^\top a}$.

Mediante transformaciones Householder sucesivas, se puede introducir ceros columna por columna, debajo de la diagonal de una matriz A, para llevarla a la forma triangular. Cuando se aplica esta transformación a un vector x, se puede apreciar que:

$$Hx = \left(I - 2\frac{vv^\top}{v^\top v}\right)x = x - 2\left(\frac{v^\top x}{v^\top v}\right)v$$

Este proceso sucesivo produce una factorización de la forma:

$$H_n \dots H_1 A = \begin{bmatrix} R \\ O \end{bmatrix}$$

donde R es triangular superior.

El producto de transformaciones Householder sucesivas $H_n \dots H_1$ es también una matriz ortogonal. Por lo tanto, si se toma:

$$Q^\top = H_n \dots H_1$$
$$Q = H_1^\top \dots H_n^\top$$

Entonces:

$$A = Q \begin{bmatrix} R \\ O \end{bmatrix}$$

Esto quiere decir que al final se debe resolver el siguiente problema de mínimos cuadrados triangular equivalente:

$$\begin{bmatrix} R \\ O \end{bmatrix} x \approx Q^\top b$$

mediante el Método de Sustitución Sucesiva hacia atrás.

3. Resultados de las simulaciones

3.1. Ejemplo 1

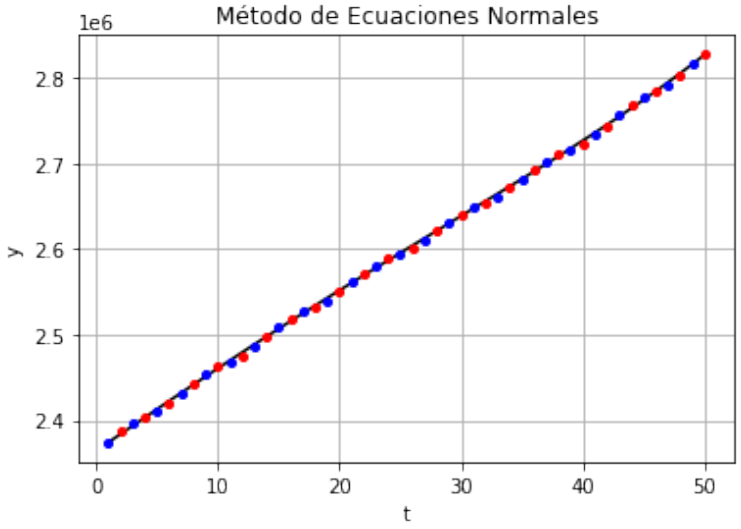
El conjunto de datos fue tomado de GitHub y describe la cantidad de muertos al día por Covid-19 desde el 22 de enero de 2020. Como el repositorio se actualiza diariamente se tomaron los datos hasta el 1 de abril de 2021.

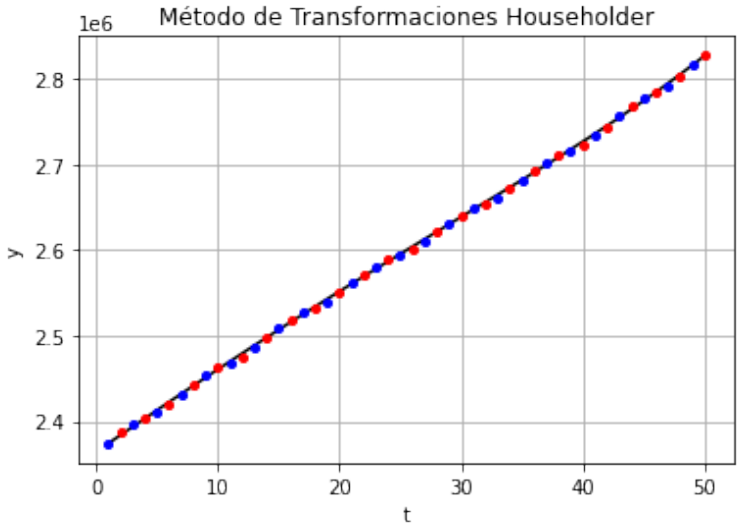
3.1.1. Parámetros x

Los parámetros calculados (con $n = 5$) por los métodos son los que se muestran en la siguiente tabla (aproximadamente):

x	Ecuaciones Normales	Transformaciones Householder
x_1	2364666.37944	2364666.37944
x_2	9748.75733	9748.75733
x_3	-4.34688	-4.34688
x_4	-1.1906	-1.1906
x_5	0.02146	0.02146

En las siguientes gráficas se muestran los datos de entrenamiento (azul) y validación (rojo), además de la gráfica de la función con los parámetros calculados, para ambos métodos.



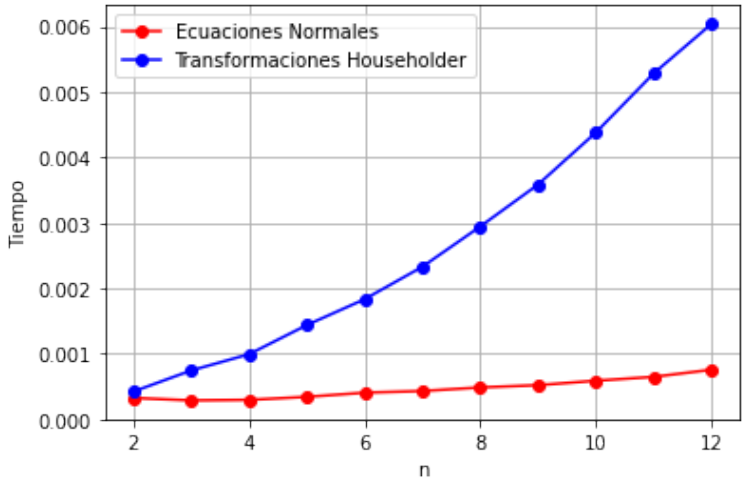


3.1.2. Tiempo de cómputo

El tiempo de ejecución (en segundos) para cada método según n se muestra en la siguiente tabla:

n	Ecuaciones Normales	Transformaciones Householder
2	0.0003242493	0.0004220009
3	0.0002865791	0.0007443428
4	0.0002949238	0.0009922981
5	0.0003409386	0.0014381409
6	0.0004045963	0.0018336773
7	0.0004303455	0.0023300648
8	0.0004847050	0.0029447079
9	0.0005199909	0.0035860538
10	0.0005874634	0.0043816566
11	0.0006465912	0.0052850246
12	0.0007555485	0.0060405731

Que se puede observar mejor en la siguiente gráfica, donde la línea roja representa el tiempo que gastó el Método de Ecuaciones Normales, y la línea azul el Método de Transformaciones Householder:

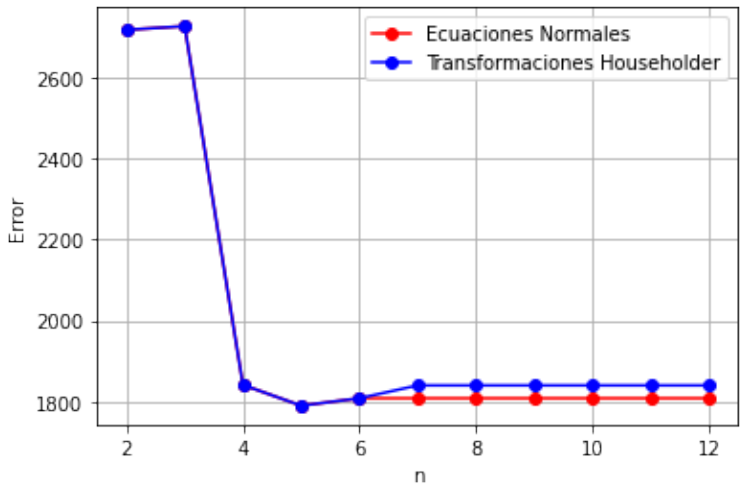


3.1.3. Exactitud de los algoritmos

La exactitud de los algoritmos se obtuvo mediante el error absoluto promedio ($\bar{e}$) y su respectiva desviación estándar (σ_e). En la siguiente tabla se encuentra el correspondiente error y desviación para cada n con cada método (EN: Ecuaciones Normales, TH: Transformaciones Householder):

n	$\bar{\mathbf{e}}$ **(EN)**	$\sigma_{\mathbf{e}}$ **(EN)**	$\bar{\mathbf{e}}$ **(TH)**	$\sigma_{\mathbf{e}}$ **(TH)**
2	2717.32943	1772.53392	2717.32943	1772.53392
3	2726.36026	1857.49014	2726.36026	1857.49014
4	1841.10552	1139.31085	1841.10552	1139.31085
5	1789.74768	941.22977	1789.74768	941.22977
6	1808.11779	879.34613	1808.11779	879.34613
7	1808.11779	879.34613	1839.81187	843.12879
8	1808.11779	879.34613	1839.81187	843.12879
9	1808.11779	879.34613	1839.81187	843.12879
10	1808.11779	879.34613	1839.81187	843.12879
11	1808.11779	879.34613	1839.81187	843.12879
12	1808.11779	879.34613	1839.81187	843.12879

Que se puede observar mejor en la siguiente gráfica, donde la línea roja representa el error del Método de Ecuaciones Normales, y la línea azul el del Método de Transformaciones Householder:


Ecuaciones Normales
Transformaciones Householder
Error
n
2600
2400
2200
2000
1800
2
4
6
8
10
12

3.2. Ejemplo 2

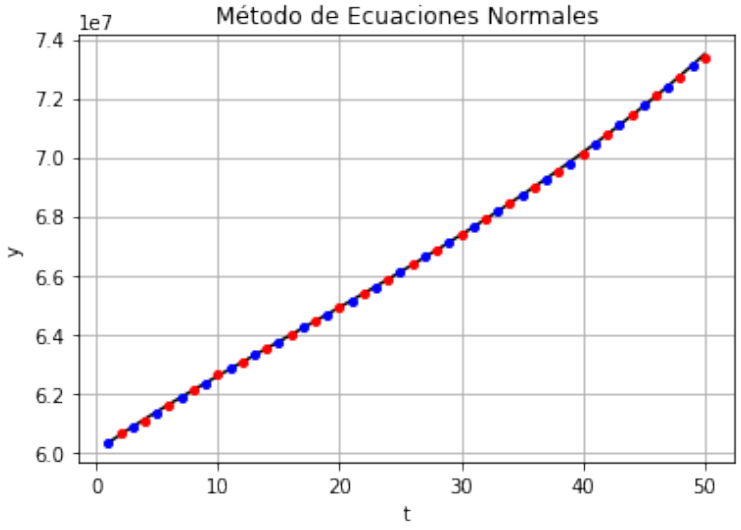
El conjunto de datos fue tomado de GitHub y describe la cantidad de recuperados al día por Covid-19 desde el 22 de enero de 2020. Como el repositorio se actualiza diariamente se tomaron los datos hasta el 1 de abril de 2021.

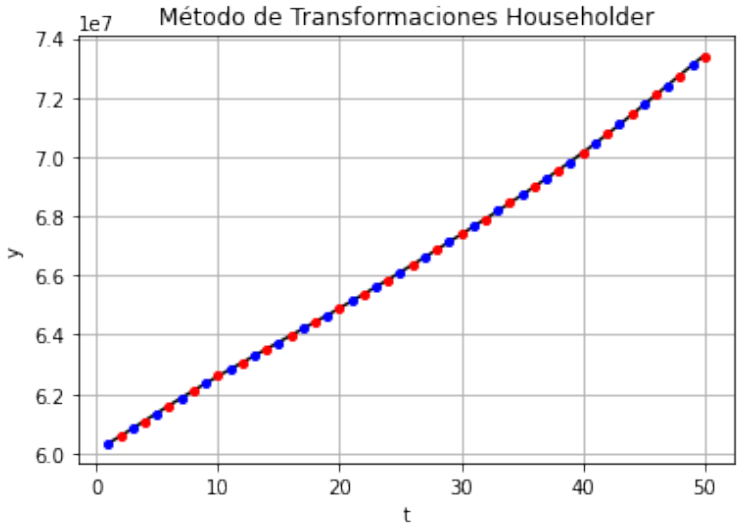
3.2.1. Parámetros x

Los parámetros calculados (con $n = 8$) por los métodos son los que se muestran en la siguiente tabla (aproximadamente):

x	Ecuaciones Normales	Transformaciones Householder
x_1	60040907.2894	60102936.13371
x_2	285667.29052	223506.97499
x_3	-4545.33081	11624.3867
x_4	163.46677	-1605.30501
x_5	-2.45019	94.20275
x_6	0.0173	-2.76107
x_7	0.0	0.04016
x_8	0.0	-0.00023

En las siguientes gráficas se muestran los datos de entrenamiento (azul) y validación (rojo), además de la gráfica de la función con los parámetros calculados, para ambos métodos.



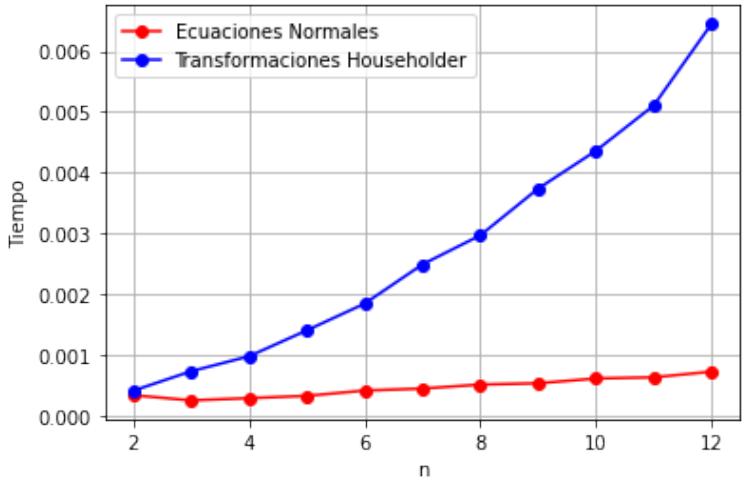


3.2.2. Tiempo de cómputo

El tiempo de ejecución (en segundos) para cada método según n se muestra en la siguiente tabla:

n	Ecuaciones Normales	Transformaciones Householder
2	0.0003383160	0.0004136562
3	0.0002529621	0.0007331371
4	0.0002901554	0.0009787083
5	0.0003275871	0.0014057159
6	0.0004153252	0.0018455982
7	0.0004479885	0.0024905205
8	0.0005140305	0.0029685497
9	0.0005350113	0.0037364960
10	0.0006136894	0.0043594837
11	0.0006330013	0.0051045418
12	0.0007259846	0.0064468384

Que se puede observar mejor en la siguiente gráfica, donde la línea roja representa el tiempo que gastó el Método de Ecuaciones Normales, y la línea azul el Método de Transformaciones Householder:

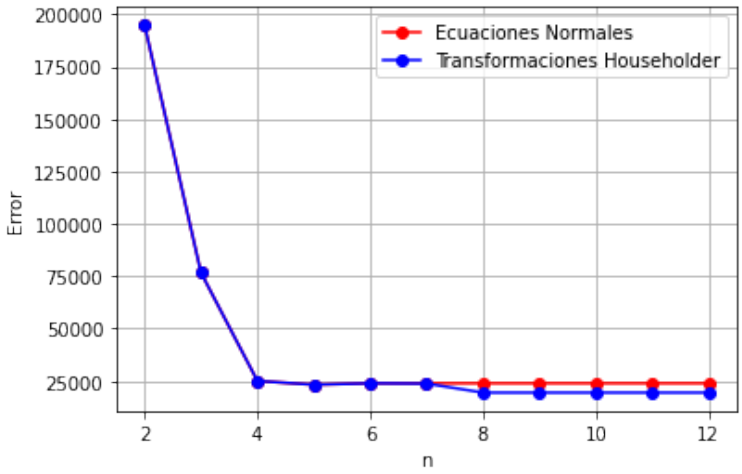


3.2.3. Exactitud de los algoritmos

La exactitud de los algoritmos se obtuvo mediante el error absoluto promedio ($\bar{e}$) y su respectiva desviación estándar (σ_e). En la siguiente tabla se encuentra el correspondiente error y desviación para cada n con cada método (EN: Ecuaciones Normales, TH: Transformaciones Householder):

n	**ē (EN)**	**σ_e (EN)**	**ē (TH)**	**σ_e (TH)**
2	194815.59586	127237.37403	194815.59586	127237.37403
3	76649.11487	39510.27271	76649.11487	39510.27271
4	25026.31218	19706.57300	25026.31218	19706.57300
5	23260.41835	16451.97751	23260.41835	16451.97751
6	23915.24482	20913.16726	23915.24482	20913.16726
7	23915.24482	20913.16726	23768.18819	20779.85899
8	23915.24482	20913.16726	19460.40261	14285.49092
9	23915.24482	20913.16726	19460.40261	14285.49092
10	23915.24482	20913.16726	19460.40261	14285.49092
11	23915.24482	20913.16726	19460.40261	14285.49092
12	23915.24482	20913.16726	19460.40261	14285.49092

Que se puede observar mejor en la siguiente gráfica, donde la línea roja representa el error del Método de Ecuaciones Normales, y la línea azul el del Método de Transformaciones Householder:


Ecuaciones Normales
Transformaciones Householder
Error
n
200000
175000
150000
125000
100000
75000
50000
25000
2
4
6
8
10
12

4. Discusión y análisis

Al usar los métodos implementados en conjuntos de datos reales, se pueden establecer las siguientes consideraciones.

4.1. Resultados de los métodos

Los resultados de los métodos fueron similares (no necesariamente iguales) hasta $n = 6$ en el ejemplo 1 y hasta $n = 7$ en el ejemplo 2. En el ejemplo 1 el Método de Ecuaciones Normales obtuvo mejor desempeño (menor error) a partir de $n = 7$, mientras que en el ejemplo 2 el Método de Transformaciones Householder obtuvo mejor desempeño a partir de $n = 8$.

4.2. Límite de los algoritmos

Para ambos ejemplos, el n más alto hasta donde el Método de Ecuaciones Normales logró dar la respuesta es $n = 12$, esto gracias a que se usó el Método de Gauss para resolver el sistema $A^\top Ax = A^\top b$ en vez de usar la descomposición de Cholesky, ya que con este último el límite era $n = 6$. Como ya se había mencionado, se realizaron los análisis para $n = 2, \ldots, 12$ para poder comparar los métodos en cada ejemplo.

4.3. Complejidad computacional

La complejidad computacional del Método de Transformaciones Householder es más alta que la del Método de Ecuaciones Normales, ya que la primera debe hacer varios cálculos que implican ciclos anidados de $n \times n \times m$, mientras que en la segunda se aproxima una complejidad de $n \times n \times n$. Como $m \geq n$ (en el ejemplo 1 $m = 25$ y $n = 5$, en el ejemplo 2 $m = 25$ y $n = 8$) entonces el Método de Transformaciones Householder tiene complejidad computacional más alta que la del Método de Ecuaciones Normales.

4.4. Mejores valores de n

En el ejemplo 1, el valor de n para el cual se obtuvo el mejor ajuste polinomial (menor error) fue $n = 5$ con ambos métodos, mientras que en el ejemplo 2 fue $n = 8, \ldots, 12$ con el Método de Transformaciones Householder. Cabe aclarar que aunque el Método de Transformaciones Householder posiblemente provea un mejor ajuste con un $n \geq 12$, para el alcance de este laboratorio sólo se analizó hasta $n = 12$ que es el límite del Método de Ecuaciones Normales.

5. Conclusiones

Se logró comprender las diferentes formas de obtener los parámetros de una función para un ajuste polinomial y la importancia de los métodos aprendidos para diversas aplicaciones.

Es relevante analizar los diferentes métodos definidos para un problema, tanto en exactitud como en complejidad, ya que esto permite decidir, según el problema y los datos, cuál es el más apropiado para realizar los cálculos.

Al usar la computación para resolver problemas matemáticos se puede profundizar en el comportamiento de los métodos tanto a nivel de exactitud como a nivel de complejidad, con lo que se logra aprender más sobre ellos e identificar en qué caso usarlos y de qué manera se puede aprovechar mejor.

Es importante experimentar con las diferentes formas de solucionar un problema matemático, en especial si se implementa en un computador, pues permite obtener resultados de una manera ágil una vez se haya construido, con lo que se facilita el uso del método cuando ya se ha comprendido o cuando comprenderlo no es el objetivo principal.

6. Referencias

- Material del curso, disponible en BlackBoard
- Bornemann, F., 2016. *Numerical linear algebra*. 1st ed. *Simson, W.*
- Mathews, J., Fink, K., Fernández Carrión, A. & Contreras Márquez, M., 2011. *Métodos Numéricos con MATLAB*. 3rd ed. Madrid: *Pearson Prentice Hall*.
- Repositorio en GitHub de los conjuntos de datos
- Librería Numpy
- Librería Pyplot (Matplotlib)
- Librería Time
- Librería Pandas